REMARKS

Claims 1-7, 9-23, and 27-34 are pending. Claims 1, 13, 18, and 27 are in independent form.

CLAIMS 18 AND 27

In the action mailed May 31, 2005, claims 18 and 27 were rejected under 35 U.S.C. § 103(a) as obvious over U.S. Patent No. 6.094,729 to Mann (hereinafter "Mann") and U.S. Patent Publication No. 2003/0115424 to Bachand et al. (hereinafter "Bachand").

Claim 18 relates to a method that includes performing a trace operation and performing a compression operation. The trace operation includes storing an address pair corresponding to a loop in fetched instructions in the trace buffer. The compression operation includes comparing a stored address pair to a new address pair, and setting a least significant bit of an address in the stored address pair in response to the new address pair matching a stored address pair.

Claim 27 relates to an apparatus. The apparatus includes instructions residing on a machine-readable medium for use in a trace buffer. The instructions are operable to a machine to perform a trace operation and perform a compression operation. The trace operation includes storing an address pair corresponding to a loop in fetched instructions in the trace buffer. The compression operation includes comparing the stored address pair to a new address pair in the fetched instructions, and setting a least significant bit of an address in the stored address pair in response to the new address pair matching the stored address pair.

The rejection contends that one of ordinary skill could use Bachand's snoop blocking technique in Mann's debug interface to arrive at the subject matter of claims 18 and 27.

Applicant respectfully disagrees.

Mann's debug interface includes a trace buffer 200 that stores entries indicative of the order in which instructions are performed by a processor. *See* Mann, col. 8, line 10-13. Since trace buffer 200 has a limited storage capacity, Mann describes that trace compression is used to limit the amount of data stored at trace buffer 200. *See* Mann, col. 18, line 10-12.

Mann's trace compression relies upon screening of the information that is reported in the trace data to limit the amount of stored data. In particular, Mann prefers that the only instructions which are reported are those which result from disruptions in the program flow. *See* Mann, col. 18, line 19-20. Such instructions are those in which the target address is data dependent, and do not include every call or branch. *See, e.g.,* Mann, col. 18, line 20-25, col. 18, line 28-31, col. 18, line

33-35, col. 18, line 37-38, and col. 18, line 28-31. As a result of this screening, few of Mann's trace entries contain address values. *See, e.g.*, Mann, col. 19, line 21. Rather, many of Mann's trace entries contain individual bit indicators, as illustrated in FIG. 6A of Mann.

Since Mann describes that trace compression is to be achieved on the basis of disruptions to the program flow, Applicant submits that one of ordinary skill would not be motivated to compare a stored address pair in the trace buffer to a new address pair to perform a compression operation, as claimed. For example, few of Mann's trace entries contain address values that would be available for comparison.

Further, one of ordinary skill would not be motivated to use those few trace entries for purposes of trace compression. This would involve a departure from Mann's basis for trace compression, namely, disruptions to the program flow. Disruptions to the program flow are departures from the expected program flow, whereas the claimed stored address pairs correspond to a loop in fetched instructions, i.e., during a previous operation. These are separate items and would not be considered the same by one of ordinary skill in the art.

Further, even if one were turn to the few of Mann's trace entries that do contain address values in performing trace compression, Bachand neither describes nor suggests comparing a stored address pair to a new address pair in fetched instructions, where the stored address pair corresponds to a loop in fetched instructions.

Bachand is directed toward maintaining cache coherency in a multi-agent architecture. *See* Bachand, para. [0002]. In Bachand's system, an external transaction queue compares the address of the newly requested data with addresses of pending posted transaction data using match detection logic 244. *See* Bachand, para. [0036-0037].

Applicant submits that Bachand's newly requested data address is neither a new address pair in fetched instructions nor a stored address pair corresponding to a loop in fetched instructions, as recited. Indeed, Bachand's newly requested data has only been newly requested, and not fetched.

Further, pending posted transaction data addresses are neither a new address pair in fetched instructions nor a stored address pair corresponding to a loop in fetched instructions, as recited. Indeed, Bachand's pending posted transaction data addresses have pending posted transactions. Since the transactions are pending, the data has not been fetched.

Since neither Bachand's newly requested data nor Bachand's pending transaction data have been accessed when they are compared, they cannot be address pairs in fetched instructions. Thus, Bachand neither describes nor suggests comparing a stored address pair to a new address pair in fetched instructions, where the stored address pair corresponds to a loop in fetched instructions.

Accordingly, claims 18 and 27, and the claims dependent therefrom, are not obvious over Mann and Bachand, individually or in combination.

CLAIM 1

Claim 1 was rejected under 35 U.S.C. § 103(a) as obvious over Mann, U.S. Patent No. 5,553,010 to Tanihara et al. (hereinafter "Tanihara"), and Bachand.

Claim 1 relates to a trace buffer circuit. The circuit includes a first comparator and a second comparator. The first comparator is to compare a new branch target address corresponding to a loop in a first holding register to a stored branch target address in a first end register. The second comparator is to compare a new branch source address corresponding to the loop in a second holding register to a stored branch source address in a first adjacent register.

The rejection of claim 1 contends that one of ordinary skill could use Bachand's snoop blocking technique in Mann's debug interface with Tanihara's double shifting system to arrive at the subject matter of claim 1.

Applicant respectfully disagrees.

As discussed above, Mann's trace compression relies upon screening of the information that is reported in the trace data on the basis of disruptions in the program flow. As a result of this screening, few of Mann's trace entries contain address values.

Since Mann describes that trace compression is to be achieved on the basis of disruptions to the program flow, Applicant submits that one of ordinary skill would not be motivated to compare a new branch target address with a stored branch target address in a first comparator or to compare a new branch source address with a stored branch source address in a second comparator, as claimed. To begin with, few of Mann's trace entries contain address values that would be available for comparison.

Further, there is no reason to believe that one of ordinary skill would turn to those few trace entries for purposes of trace compression. This would involve a departure from Mann's basis for trace compression, namely, disruptions to the program flow.

Moreover, even if one were to turn to the few of Mann's trace entries that do contain address values in performing trace compression, Bachand neither describes nor suggests comparing branch target addresses in a first comparator or comparing branch source addresses in a second comparator, as claimed.

In particular, Bachand includes a single comparator. That single comparator compares the address of newly requested data with addresses of pending posted transaction data. *See* Bachand, para. [0036-0037]. These compared addresses are neither branch target addresses nor branch source addresses. Rather, they are simply generic data addresses.

The rejection also contends that Bachand's snoop queue 250 constitutes a first holding register and Bachand's external queue 240 constitutes a second holding register. However, Bachand describes that the address of newly requested data is provided by Bachand's snoop queue (i.e., the alleged first holding register) to external transaction queue (i.e., the alleged second holding register) that includes the addresses of pending, posted transaction data for comparison by match detection logic 244. *See* Bachand, para. [0033]. Bachand thus describes that information in the alleged first holding register is to be compared with information in the alleged second holding register.

However, in claim 1, the information in the first holding register is <u>not</u> compared with the information in the second holding register. Rather, the new branch target address in the *first holding register* is compared to a stored branch target address in the *first end register*, and the new branch source address *in the second holding register* is compared to a stored branch source address in the *first adjacent register*.

Thus, Bachand neither describes nor suggests the comparison of any data in a first holding register with data in a first end register. Also, Bachand neither describes nor suggests the comparison of any data in a second holding register with data in a first adjacent register.

Tanihara does nothing to remedy these deficiencies. Tanihara describes a data shifting circuit of a central processor and has nothing to do with either trace buffer circuits or trace operations.

Accordingly, claim 1, and the claims dependent therefrom, are not obvious over Mann, Tanihara, and Bachand, individually or in combination.

<u>CLAIM 13</u>

Claim 13 was rejected under 35 U.S.C. § 103(a) as obvious over Mann and Tanihara.

Claim 13 relates to a pipelined processor. The processor includes a trace buffer circuit. The trace buffer circuit includes a plurality of interconnected registers, including a first end register to input and output addresses of fetched instructions during a trace operation, a second end register, and a plurality of middle registers connected between said first end register and said second end register, a write path to shift an instruction address in one of said plurality of interconnected registers by two registers toward the second end register on a write operation, and a read path to shift the instruction address by one register toward the first end register on a read operation.

To begin with, neither Mann nor Tanihara describes or suggests a pipelined processor. On this basis alone, claim 13, and the claims dependent therefrom, are not obvious over Mann and Tanihara, individually or in combination.

Further, the rejection contends that it would have been obvious to one of ordinary skill to use Tanihara's double data width shifting in Mann's debug interface to accommodate inputs that were "double the size of one register."

This contention neglects the fact that Mann's trace entries are specified to be the 20 bits in length. *See, e.g.*, FIGS. 6A-6G and the written descriptions thereof. Indeed, Mann describes that otherwise complete data is to be divided in order to accommodate this 20 bit trace entry length. *See, e.g.*, FIG. 6B (illustrating that the complete 32 bits of an EIP target logical address is to be divided into a high 16 bits and a low 16 bits to maintain the 20 bit trace entry length).

Without the threat of double sized inputs, there is no reason to believe that one of ordinary skill would be motivated to add a write path to shift an instruction address in one of said plurality of interconnected registers by two registers on a write operation, as claimed. Indeed, Mann can fairly be seen as teaching away from shifting an instruction address by two registers, given that Mann describes that care is to be taken to ensure that the trace entries in his FIFO trace buffer 200 are all to be the same length.

Accordingly, claim 13, and the claims dependent therefrom, are not obvious over Mann and Tanihara, individually or in combination.

Applicant asks that all claims be allowed. No fees are believed due at this time. Please apply any charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

BY
JOHN F. CONROY
REG. NO. 45,485

Date: August 8, 2005

Scott C. Harris
Reg. No. 32,030
Attorney for Intel Corporation

Fish & Richardson P.C.
PTO Customer No. 20985
12390 El Camino Real
San Diego, California 92130
(858) 678-5070 telephone
(858) 678-5099 facsimile

10537775.doc